

1.2 SOFTWARE & SOFTWARE DEVELOPMENT · 1.2.1

Interrupts & scheduling — Mark scheme

33 marks · spec 1.2.1(c)(d)

AO key: AO1 = knowledge & understanding · AO2 = application · AO3 = reasoned judgements. Accept any valid alternative; do not award the same point twice.

Q	ANSWER	AO	MARKS
1(a)	An interrupt. (1)	AO1	1
1(b)	1 mark per source, max 2: hardware (device) / software (program) / timer (clock). (2)	AO1	2
1(c)	Interrupt service routine (ISR). (1)	AO1	1

Q	ANSWER	AO	MARKS
2(a)	1 mark per point, max 4: <ul style="list-style-type: none"> • finish the current instruction / fetch-decode-execute cycle (1) • save the current state (registers and program counter) to the stack (1) • load and run the relevant interrupt service routine (ISR) (1) • restore the saved state and resume the interrupted program (1) 	AO1	4
2(b)	<ul style="list-style-type: none"> • so more urgent interrupts are dealt with first (1) • a higher-priority interrupt can interrupt a lower-priority one; lower ones wait (1) 	AO2	2

Q	ANSWER	AO	MARKS
3(a)	1 mark per point, max 3: <ul style="list-style-type: none"> • a single CPU can run only one process at a time but many need it (1) • scheduling decides which process runs and for how long (1) • to maximise throughput / be fair / give good response times / keep resources busy (1) 	AO1	3
3(b)	1 mark per point, max 3: <ul style="list-style-type: none"> • each process is given a fixed time slice / quantum in turn (1) • processes are taken on a FIFO basis (1) • if a process does not finish in its slice it goes to the back of the queue (1) 	AO1	3

Q	ANSWER	AO	MARKS
4(a)	<ul style="list-style-type: none"> • jobs run strictly in arrival order with no priorities (1) • a long job at the front makes all jobs behind it wait (poor response time) (1) 	AO2	2
4(b)	The process with the smallest estimated total run time is run next (1); (requires an estimate of each job's length) (1). (2)	AO1	2
4(c)	A job whose run time is known/predictable, e.g. a regular batch job such as payroll. (1)	AO2	1

Q	ANSWER	AO	MARKS
5(a)	1 mark per point, max 3: <ul style="list-style-type: none"> • uses several queues with different priorities (1) • jobs move between queues depending on how much CPU time they use (1) • gives preference to short / I/O-bound jobs to keep devices busy (1) 	AO1	3
5(b)	Any one: maximise throughput / fairness / acceptable response time / keep resources busy. (1)	AO1	1

Q	LEVELS-OF-RESPONSE MARK SCHEME	AO	MARKS										
6	Mark using the levels descriptors below. AO1 (knowledge of algorithms), AO2 (application to the mixed workload), AO3 (justified recommendation). <table border="1" data-bbox="256 647 1197 1111"> <thead> <tr> <th>LEVEL</th> <th>DESCRIPTOR</th> </tr> </thead> <tbody> <tr> <td>Level 3 (7–8)</td> <td>Discusses suitable algorithms for both the interactive and the batch workload, clearly applied to the server, leading to a justified recommendation (e.g. round robin for interactive, SJF/ SRT for overnight batch). Well structured.</td> </tr> <tr> <td>Level 2 (4–6)</td> <td>Considers more than one algorithm with some application to the context; a recommendation is made with partial justification.</td> </tr> <tr> <td>Level 1 (1–3)</td> <td>Basic points about one or more algorithms with little application or justification.</td> </tr> <tr> <td>0</td> <td>Nothing creditworthy.</td> </tr> </tbody> </table> <p>Indicative content (credit any valid point):</p> <ul style="list-style-type: none"> • Interactive student programs need fast, fair response → round robin with a suitable quantum guarantees regular turns. • Long overnight batch jobs are not time-critical and their length is often predictable → shortest job first / shortest remaining time, or run them when interactive load is low. • Multi-level feedback queues could combine both, giving short/interactive jobs priority while still progressing long jobs. • Justified conclusion: e.g. round robin during the day, batch scheduling overnight, or MLFQ to balance both. 	LEVEL	DESCRIPTOR	Level 3 (7–8)	Discusses suitable algorithms for both the interactive and the batch workload, clearly applied to the server, leading to a justified recommendation (e.g. round robin for interactive, SJF/ SRT for overnight batch). Well structured.	Level 2 (4–6)	Considers more than one algorithm with some application to the context; a recommendation is made with partial justification.	Level 1 (1–3)	Basic points about one or more algorithms with little application or justification.	0	Nothing creditworthy.	AO1 ×3 AO2 ×3 AO3 ×2	8
LEVEL	DESCRIPTOR												
Level 3 (7–8)	Discusses suitable algorithms for both the interactive and the batch workload, clearly applied to the server, leading to a justified recommendation (e.g. round robin for interactive, SJF/ SRT for overnight batch). Well structured.												
Level 2 (4–6)	Considers more than one algorithm with some application to the context; a recommendation is made with partial justification.												
Level 1 (1–3)	Basic points about one or more algorithms with little application or justification.												
0	Nothing creditworthy.												

Total for paper: 33 marks